

performing the request, and terminating the connection. The HTTP server does not maintain any state about the connection once it has been terminated. HTTP is, therefore, a stateless application protocol. That is, a client can make several requests of an HTTP server, but each individual request is treated independent of any other request. The server has no recollection of any previous request.

[0035] Internet Applications and Programming Environment

[0036] Although Internet applets or application can be developed and executed in any Internet programming environment, they have been described here, by way of example, in connection with the Java language and programming environment.

[0037] Java is an object-oriented programming language with each program comprising one or more object classes and interfaces. Unlike many programming languages in which a program is compiled into machine-dependent, executable program code, classes written in the Java programming language are compiled into machine independent bytecode class files. Each class contains code and data in a platform-independent format called the class file format. A bytecode includes a code that identifies an instruction (an opcode) and none or more operands to be used in executing the instruction. The computer system acting as the execution vehicle contains a program called a virtual machine, which is responsible for executing the code (i.e., bytecode) in Java programming language class files.

[0038] Client applications may be designed as standalone Java applications, or as Java "applets" which are identified by an applet tag in an HTML document, and loaded by a browser application. The class files associated with an application or applet may be stored on the local computing system, or on a server accessible over a network. Each Java programming language class file is loaded into the Java virtual machine, as needed, by the "class loader."

[0039] To provide a client with access to class files from a server on a network, a web server application is executed on the server to respond to HTTP requests containing URLs (Universal Resource Locators) to HTML documents, also referred to as "web pages." When a browser application executing on a client platform receives an HTML document (e.g., as a result of requesting an HTML document by forwarding a URL to the web server), the browser application parses the HTML and automatically initiates the download of the specified bytecode class files when it encounters an applet tag in the HTML document.

[0040] The classes of a Java applet are loaded on demand from the network (stored on a server), or from a local file system, when first referenced during the Java applet's execution. The virtual machine locates and loads each class file, parses the class file format, allocates memory for the class's various components, and links the class with other already loaded classes. This process makes the code in the class readily executable by the virtual machine. Native code, e.g., in the form of a dynamic linked library (DLL), is loaded when a Java programming language class file containing the associated native method is instantiated within the virtual machine.

[0041] A Java applet loaded from the network server is executed on the client's virtual machine. An applet has

limited permission to access the resources available on the server and other network computers. In prior art schemes, this access is typically limited to the resources available on the server where the applet is loaded from. This is because an applet retrieved from a server has a trusted status with that server. A trusted applet is one that can directly connect to a server computer. However, due to security issues, such as firewalls and the limitations associated with the browser architecture, an applet cannot directly access information on other computers on the network by which it is not trusted.

[0042] FIG. 1 is a block diagram of a client/server system connected via the Internet. Client application 101 can be an application or applet downloaded from office server 102, via the Internet connection, onto a user's home computer. Typically, applications downloaded from a server are trusted by that server. Thus, client application 101 can access the resources available on office server 102.

[0043] It may be desirable for client application 101 to access the resources available on other computers on the office network, such as office computers 103, 104, or 105. However, client application 101 may not be entrusted by those network computers. As a result it cannot directly authenticate against those computers and hence cannot access information available thereon. Current methods do not provide for any means by which an application can access resources on a network computer, via the Internet, unless it can directly authenticate against it. A method is needed to allow an application, trusted to a first computer on a network, to indirectly communicate with other network computers, via the Internet.

[0044] The Network File System (NFS)

[0045] A distributed file system provides for the sharing of files and information located on computer systems geographically separated but functionally linked together in a communications network. The Network File System or NFS was developed by Sun Microsystems and an example of a distributed file system protocol that allows a computer on a network to use the files and peripherals of another networked computer (remote computer) as if they were local to that computer. NFS operates as a client server application. An NFS server is a computer that shares its resources with other computers (NFS clients) on the network, using the NFS service.

[0046] Using NFS, a resource (i.e., software) physically linked to a NFS server may be "NFS mounted." The resource that is "NFS mounted" is accessible to all NFS clients as if the software were stored locally at each client. This is accomplished by a system of tables (mount tables) identifying the mounted resource and the appropriate path names to the NFS server where the resources resides. The NFS system can also be used to locate a designated resource so that it can be accessed from a remote location.

[0047] The NFS system generated mount table is stored as a file on each server and client in the NFS network. The mount table consists of a list of entries. When a new resource is mounted, a new entry is generated in the table automatically and is used to locate the resource whenever a request for access to that resource is made by a client.

[0048] FIG. 2 is a block diagram illustrating the components of an NFS system and the interrelationship between those components. An NFS system includes application 210